

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

До захисту допущено

В.о. завідувача кафедри

\_\_\_\_\_  
(підпис) О.В. Коваль  
(ініціали, прізвище)  
“ \_\_\_\_ ” \_\_\_\_\_ 2019р.

**ДИПЛОМНА РОБОТА**  
**на здобуття ступеня бакалавра**

з напрямку підготовки 6.050101 «Комп’ютерні науки»

на тему Оцінка ресурсів відновлюваних джерел енергії території з використанням показника сумарного питомого нормованого потенціалу

Виконала: студентка 4 курсу, групи ТМ-51

\_\_\_\_\_  
Сергійчук Алла Сергіївна

(прізвище, ім’я, по батькові)

\_\_\_\_\_  
(підпис)

Керівник к.е.н., доцент Сегеда І.В.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

\_\_\_\_\_  
(підпис)

Рецензент \_\_\_\_\_

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

\_\_\_\_\_  
(підпис)

Засвідчую, що у цій дипломній роботі немає  
запозичень з праць інших авторів без  
відповідних посилань.

Студент \_\_\_\_\_  
(підпис)

Київ – 2019

**Національний технічний університет України**  
**«Київський політехнічний інститут імені Ігоря Сікорського»**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти перший, бакалаврський

Напрямок підготовки 6.050101 «Комп'ютерні науки»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

\_\_\_\_\_ О.В. Коваль  
(підпис)

” ” \_\_\_\_\_ 2019р.

**ЗАВДАННЯ**  
**на дипломну роботу студенту**  
**Сергійчук Аллі Сергіївні**  
(прізвище, ім'я, по батькові)

1. Тема роботи Оцінка ресурсів відновлюваних джерел енергії території з використанням показника сумарного питомого нормованого потенціалу

керівник роботи к.е.н., доцент Сегеда Ірина Василівна  
(прізвище, ім'я, по батькові науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від «22» травня 2019р. №1324

2. Строк подання студентом роботи «10» червня 2019р.

3. Вихідні дані до роботи: мова програмування — С#, середовище розробки — Microsoft Visual Studio 2017

4. Зміст розрахунково-пояснювальної записки (перелік завдань, які потрібно розробити) розглянути та проаналізувати методи оцінки ресурсів відновлюваних джерел енергії території, обґрунтувати вибір моделі оцінки ресурсів відновлюваних джерел енергії території з використанням показника сумарного питомого нормованого потенціалу та алгоритм розробки програмного додатку, розробити програмне забезпечення, розробити інтерфейс користувача, зробити висновки за результатами роботи.

5. Перелік ілюстративного матеріалу

1. Блок-схема алгоритму оцінки ресурсів ВДЕ території. 2. Функціональний

принцип роботи системи. 3. Компоненти .NET Framework. 4. Модель Entity Data Model. 5. Каскадна модель розробки програмного забезпечення. 6. Взаємодія структурних блоків програмного продукту. 7. Структура модулів проекту. 8. Концептуальна схема бази даних. 9. Фрагмент таблиці “Region resources”. 10. Фрагмент таблиці “Region resources normalized”. 11. Головна форма програми. 12. Обчислення нормованих показників ВДЕ. 13. Мапа України з показниками ВДЕ для кожного регіону. 14. Районування території України за ресурсним потенціалом вітрової енергії. 15. Районування території України за ресурсним потенціалом сонячної енергії. 16. Районування території України за ресурсним потенціалом гідроенергії. 17. Районування території України за потенціалом геотермальної енергії.

6. Дата видачі завдання «10» жовтня 2018 р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітки
1.	Вивчення та аналіз задачі	12.02.2019 – 18.03.2019	
2	Розробка архітектури та загальної структури системи	27.03.2019 – 03.04.2019	
3.	Розробка структур окремих підсистем	09.04.2019 – 16.04.2019	
4.	Програмна реалізація системи	17.04.2019 – 12.05.2019	
5.	Оформлення пояснювальної записки	03.05.2019 – 06.06.2019	
6.	Захист програмного продукту	14.05.2019	
7.	Передзахист	29.05.2019	
8.	Захист	18.06.2019	

Студент

\_\_\_\_\_  
(підпис)

Сергійчук А.С.  
(прізвище та ініціали.)

Керівник роботи

\_\_\_\_\_  
(підпис)

Сегеда І.В.  
(прізвище та ініціали.)

# **АНОТАЦІЯ**

Дана робота присвячена створенню програмної системи для оцінки ресурсів відновлюваних джерел енергії території з використанням показника сумарного питомого нормованого потенціалу. Програмний продукт являє собою додаток, створений на мові програмування C# та з використанням системи керування базами даних MySQL Server. Додаток може використовуватись користувачами системи для аналізу та якісної оцінки ресурсів та потенціалу відновлюваних джерел енергії території України та проведення територіального районування за потенціалом ресурсів ВДЕ.

Обсяг пояснювальної записки складає 65 сторінок і містить у собі 17 ілюстрацій, та 15 посилань на літературу.

Ключові слова: оцінка ресурсів ВДЕ, нормалізація показників, показник сумарного нормованого потенціалу, альтернативна енергетика, відновлювальні джерела енергії.

# **ABSTRACT**

This work is devoted to the creation of a software system for assessing the resources of renewable energy sources of the territory using the indicator of the total specific normalized potential. The software product is an application created in the C # programming language and using the MySQL Server database management system. The application can be used by users of the system for analysis and qualitative assessment of the resources and potential of renewable energy sources in the territory of Ukraine and conducting territorial zoning on the potential of RES resources.

The volume of the explanatory note is 65 pages and contains 17 illustrations, and 15 references to literature.

Key words: estimation of RES resources, normalization of indicators, indicator of total normed potential, alternative energy, renewable energy sources.

# ЗМІСТ

Перелік умовних позначень, скорочень і термінів .....	7
Вступ .....	8
1. Постановка задачі розробки програмного продукту оцінки ресурсів відновлюваних джерел енергії території .....	10
2. Розгляд і аналіз проблеми оцінки ресурсів відновлюваних джерел території України .....	12
2.1 Опис вхідних даних.....	13
2.2 Опис вихідних даних .....	15
2.3 Алгоритмічна модель оцінки ресурсів ВДЕ території .....	16
3. Засоби реалізації програмної системи.....	18
3.1 Опис використаної операційної системи .....	19
3.2 Мова програмування C# .....	19
3.3 Середовище розробки Microsoft Visual Studio 2017 .....	21
3.4 Фреймворк Entity Framework .....	22
3.5 Система керування базами даних MySQL Server .....	25
4. Опис програмної реалізації .....	26
4.1 Архітектура програмного продукту .....	30
4.2 Структура проекту .....	32
4.3 Опис бази даних проекту.....	35
5. Методика роботи користувача з програмною системою .....	37
5.1 Інсталяція та системні вимог .....	37
5.2 Навігація програмної системи .....	38
6. Аналіз результатів роботи програмної системи.....	41
6.1 Районування території за ресурсним потенціалом вітрової енергії .....	41
6.2 Районування території за ресурсним потенціалом сонячної енергії .....	42
6.3 Районування території України за гідроенергетичним потенціалом .....	43
4.3 Районування території України за потенціалом геотермальної енергії .....	44

Висновки .....	45
Список використаних джерел .....	46
Додаток 1 .....	48
Додаток 2 .....	50
Додаток 3 .....	57

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

ВДЕ — відновлювальні джерела енергії.

СПНП — сумарний питомий нормований потенціал.

С# – об'єктно-орієнтована мова програмування.

Software architecture – архітектура програмного продукту. Це структура програми або обчислювальної системи, яка включає програмні компоненти, а також відносини між ними.

DAO (Data Access Object) – це об'єкт, який надає абстрактний інтерфейс бази даних.

## **ВСТУП**

Забезпеченість країни енергоносіями є запорукою стрімкого розвитку економіки та промисловості. На сьогоднішній день актуальною проблемою в Україні залишається раціональне та екологічно чисте енергоспоживання та забезпечення регіонів паливно-енергетичними ресурсами. Особливе місце займає також районування території за показником забезпечуваності енергоресурсами. Існує ряд проблем в сфері енергетики, які потребують негайного вирішення [1]:

- значні затрати на добування паливно-енергетичної сировини;
- велика кількість шкідливих викидів внаслідок використання енергоресурсів;
- нерівномірне енергозабезпечення території країни;
- застарілі технології енергетичного господарства.

Одним із напрямків розвитку енергетики є застосування екологічно чистіших способів виготовлення електричної енергії. Сьогодні все більше і більше надається перевага відновлювальним джерелам енергії (ВДЕ), які складають конкуренцію традиційним. Ресурси цих видів енергії – невичерпні, але потрібно розумно оцінити їх потенціал і проаналізувати, чи зможуть вони задовольнити наші потреби у енергоспоживанні.

Метою даної дипломної роботи є створення програмної системи, що дозволяє оцінити ресурси відновлюваних джерел енергії території України.

Розроблено програмне забезпечення на платформі Microsoft Visual Studio 2017 з використанням об'єктно-орієнтованої мови C#, та підтримкою системи керування базами даних MySQL. Програмний комплекс має ряд базових можливостей для вирішення задачі оцінки ресурсів відновлюваних джерел енергії території з використанням показника сумарного питомого нормованого потенціалу.

Пояснювальна записка містить шість розділів.

В першому розділі виконано постановку задачі оцінки ресурсів відновлюваних джерел енергії території.

В другому розділі описано пояснення математичної моделі задачі.



Розглядається методика розрахунку нормованих значень питомого потенціалу ВДЕ для кожного регіону та сумарного показника забезпеченості областей регіону ресурсами ВДЕ. Описуються вхідні та вихідні дані , а також алгоритм реалізації програмного комплексу.

В третьому розділі розглянуто методи реалізації програмного забезпечення та обґрунтовано вибір програмних засобів для реалізації програмного продукту.

Четвертий розділ містить опис компонентів, з яких складається система, детально розглянуто архітектуру системи та самих компонентів.

П'ятий розділ описує методику роботи користувача з програмною системою, установкою програмного продукту на комп'ютер користувача, сценарії роботи з ним, встановленням мінімальних системних вимог для його оптимального виконання.

В шостому розділі проведено оцінку ресурсів ВДЕ і проведено територіальне розподілення території України на основі потенціалу цих ресурсів.

# **1. ПОСТАНОВКА ЗАДАЧІ РОЗРОБКИ ПРОГРАМНОГО ПРОДУКТУ ОЦІНКИ РЕСУРСІВ ВІДНОВЛЮВАНИХ ДЖЕРЕЛ ЕНЕРГІЇ ТЕРИТОРІЇ**

Метою дипломної роботи є створення програмної системи, що дозволяє оцінити ресурси відновлюваних джерел енергії території України.

Потенціал природних ресурсів неабияк грає роль у формуванні територіальної структури господарства. Можна зазначити, що природно-ресурсне районування є базовим для подальшого проведення економічного регіонального планування держави.

Проблема районування території залишається відкритою вже багато десятиліть в різних напрямках і галузях. Особливе значення мають ідеї Н. Н. Баранського про економічне районування на основі природних, трудових, технічних та інших ресурсів, необхідних для комплексного розвитку кожного району та всієї території загалом. Ще в 20-х роках ХХ Барановський заклав цю ідею у своїх працях, а саме у вітчизняній економічній географії [2].

Значний внесок в теорію природно-ресурсного районування вніс А.А Мінц. У своїх роботах він зазначив необхідність використання питомих величин, розрахованих на одиницю площі або на душу одиницю населення при економічній оцінці ресурсного потенціалу території [3]. На сьогоднішній день при вирішенні питання територіального планування, у економічній оцінці територій спостерігається тенденція переходу від грошового еквіваленту, до порівняння відносних величин (шляхом різних математичних обрахунків і методів).

Для ресурсного районування ВДЕ доцільно використовувати в якості територіальних одиниць області адміністративного поділу регіонів. При аналізі можливостей і перспектив використання ВДЕ і оцінці їх майбутньої ролі в енергокомплексі держави, є важливим розглядати не кожен ресурс окремо, а проводити типологію районів, на основі характеристики рівня забезпеченості ресурсами ВДЕ, що мають найбільше значення певного типу відновлюваної енергії.

Інформаційно-аналітична система оцінки ресурсів ВДЕ для кожного регіону України являє собою програмне забезпечення з картографічною і цифровою інформацією, для визначення потенціалу відновлюваних джерел енергії.

В якості основних показників було обрано такі види ВДЕ:

- сонячна енергія ;
- енергія води;
- вітрова енергія;
- геотермальна енергія (природне тепло Землі).

Користувачами даної програмної системи можуть бути спеціалісти міністерства енергетики, спеціалісти територіального районування, інженери з проектування енергетичних процесів та систем, фахівці підприємств з енергозабезпечення , а також зацікавлені спеціалісти інших профілів.

## 2. РОЗГЛЯД І АНАЛІЗ ПРОБЛЕМИ ОЦІНКИ РЕСУРСІВ ВІДНОВЛЮВАНИХ ДЖЕРЕЛ ТЕРИТОРІЇ УКРАЇНИ

Очевидно, що існує потреба в поглибленому дослідженні та якісному аналізі проблеми оцінки ресурсів відновлюваних джерел території.

Існує цілий ряд методів і моделей для вирішення цієї проблеми.

Оцінку забезпеченості регіонів ресурсами ВДЕ вважаємо за доцільне проводити в залежності від кількості жителів - споживачів енергії на одиницю площі.

Для комплексної оцінки було обрано метод розрахунку нормованих показників різних видів ВДЕ та показника сумарного питомого нормованого потенціалу для кожного регіону.

Нормування дає можливість визначити відхилення всіх показників від найменших до найбільших значень і тим самим зрівняти їх. Нормалізація показників дозволяє усунути кількісні співвідношення між значеннями різних видів ресурсів для досліджуваних територіальних одиниць, що є основою для якісного порівняння ресурсів ВДЕ різного типу і між різними регіонами.

Нормовані значення питомого потенціалу ресурсів розраховуються за формулою [4]:

$$a_{ij} = \frac{x_{ij} + x_j^0}{x_j^{max} - x_j^0} \quad (2.1)$$

Де

$i=1,2,3...n$ ;

$j=1,2,3...m$ ;

$n$  - кількість територіальних одиниць;

$m$  - кількість показників потенціалу ресурсів ( $x$ );

$x_j^0$  – поточне значення показника ресурсів певного типу ВДЕ регіону;

$x_j^0$  – найменше значення (по кожному показнику ресурсів) із всіх регіонів;

$x_j^{max}$  – найбільше значення показника.

Важлива також характеристика загальної забезпеченості територіальних одиниць ресурсами альтернативних джерел енергії. Для цього визначається сумарне значення по всім питомим нормованим потенціалам ресурсів по окремим видам ВДЕ.

Сумарний питомий нормований потенціал розраховується за формулою:

$$S_i = \sum_{j=1}^m \frac{x_{ij} + x_j^0}{x_j^{max} - x_j^0} \quad (2.2)$$

$S_i$  - Сумарний питомий нормований потенціал (СПНП).

Значення СПНП можна розділити на наступні рівні потенціалу ресурсів ВДЕ:

- дуже високий (СПНП > 2,5);
- високий (СПНП 2,0-2,5);
- середній (СПНП 1,0-2,0);
- низький (СПНП < 1,0).

Найбільші значення СПНП свідчать про те, що регіон має досить високий потенціал одночасно декількох видів відновлюваних джерел енергії.

Даний метод оцінки ресурсів ВДЕ дозволяє найбільш правильно проаналізувати показники забезпеченості територій ресурсами відновлюваних джерел енергії, та може слугувати для територіального районування в залежності від значень отриманих показників.

## 2.1 Опис вхідних даних

Вхідними даними до розробленого програмного забезпечення є наступні показники

ВДЕ по кожному регіону України:

- кількість сонячної радіації на одиницю площі;
- середня швидкість вітру на висоті 10 м;
- показник гідроенергетичних ресурсів в регіонах;

– геотермальний показник рівня енергії Землі.

Статистичні метеорологічні дані показників зазначені у таблиці 2.1.

Таблиця 2.1. Показники ресурсів ВДЕ для кожної області України [5].

Область	Кількість сонячної радіації, кВтхгод/кв.м	Швидкість вітру, м/с	Показник гідроенергії, ГВт	Геотермальний показник, кВтхгод
<u>Автономна Республіка Крим</u>	0.38	8.7	1.11	0.14
<u>Вінницька область</u>	0.21	3.2	0.31	0.09
<u>Волинська область</u>	0.18	4.2	0.24	0.03
<u>Дніпропетровська область</u>	0.27	5.1	0.38	0.02
<u>Донецька область</u>	0.27	5.27	0.32	0.05
<u>Житомирська область</u>	0.22	4.6	0.26	0.07
<u>Закарпатська область</u>	0.17	3.6	1.02	0.12
<u>Запорізька область</u>	0.24	5.3	0.97	0.03
<u>Івано- Франківська область</u>	0.19	3.4	0.41	0.14
<u>Київська область</u>	0.26	4.6	0.57	0.04
<u>Кіровоградська область</u>	0.82	4.2	0.24	0.03
<u>Луганська область</u>	0.74	5.1	0.38	0.02
<u>Львівська область</u>	0.27	5.27	0.32	0.05
<u>Миколаївська область</u>	1.05	4.6	0.26	0.11

<u>Одеська область</u>	1.51	6.6	1.02	0.13
<u>Полтавська область</u>	0.24	5.3	0.97	0.03
<u>Рівненська область</u>	0.22	4.6	0.26	0.07
<u>Сумська область</u>	0.87	7.6	1.02	0.04
<u>Тернопільська область</u>	0.24	5.3	0.97	0.03
<u>Харківська область</u>	0.36	6.4	0.41	0.07
<u>Херсонська область</u>	1.12	5.7	0.71	0.05
<u>Хмельницька область</u>	0.22	4.4	0.26	0.03
<u>Черкаська область</u>	0.24	5.9	0.97	0.03
<u>Чернівецька область</u>	0.17	4.1	0.3	0.13
<u>Чернігівська область</u>	0.26	4.6	0.57	0.04

## 2.2 Опис вихідних даних

Вихідними даними програмного забезпечення є нормалізовані показники ресурсів ВДЕ по кожному регіону України, а саме:

- питомий нормований потенціал сонячної енергії;
- питомий нормований потенціал вітрової енергії;
- питомий нормований потенціал гідроенергії;
- питомий нормований потенціал енергії Землі;

- сумарний питомий нормований потенціал всіх видів енергії.

## **2.3 Алгоритмічна модель оцінки ресурсів ВДЕ території**

Оцінка ресурсів відновлювальних джерел енергії території з використанням показника сумарного питомого нормованого потенціалу відбувається згідно таку алгоритму:

- збір статичних даних ресурсів ВДЕ для кожної області України;
- формування таблиці бази даних, що буде містити інформацію про ці показники;
- нормалізація вхідних показників для кожного регіону по кожному типу відновлювальної енергії за допомогою формули (2.1);
- розрахунок показника сумарного питомого нормованого потенціалу для визначення загальної кількості енергії для кожного регіону на основі нормалізованих питомих показників за допомогою формули (2.2);
- формування таблиці бази даних, що буде містити інформацію про питомі нормовані показники та сумарний показник для кожного регіону;
- виведення результатів користувачеві у вигляді таблиці;
- створення енергетичної мапи України шляхом графічної інтерпретації вихідних показників у вигляді стовпчастих діаграм для кожного регіону;
- оцінка ресурсів ВДЕ території шляхом аналізу створеної енергетичної мапи.

Блок-схема даного алгоритму зображена на рисунку 2.1.

Обраний лінійний алгоритм є зручний для реалізації та не затрачає багато часу та пам'яті процесору на проведення обрахунків. Алгоритм оптимізований і захищений від можливих зациклювань.



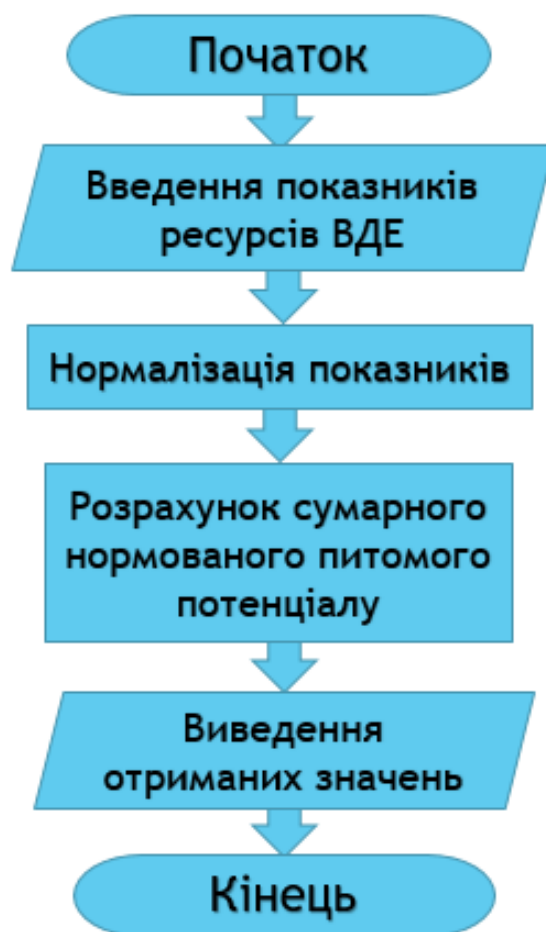


Рисунок 2.1 – Блок-схема алгоритму оцінки ресурсів ВДЕ території.

### 3. ЗАСОБИ РЕАЛІЗАЦІЇ ПРОГРАМНОЇ СИСТЕМИ

Сьогодні досить стрімко відбувається розвиток обчислювальної техніки та її компонентів. З розвитком потужності комп'ютера вдосконалюється програмне забезпечення, створюють більш ефективні та потужні алгоритми. Тому для реалізації поставленої задачі доступна велика кількість алгоритмів та засобів проектування, з яких необхідно вибрати найбільш доцільні для виконання завдання.

Для розробки програмного продукту було обрано:

- операційну систему Windows 10;
- об'єктно-орієнтовану мову програмування C#;
- середовище MS Visual Studio 2017;
- фреймворк ADO.NET Entity Framework;
- систему керування базами даних MySQL Server.

Як саме забезпечується взаємодія користувачів та вище названих системних засобів зображено на рисунку 3.1.

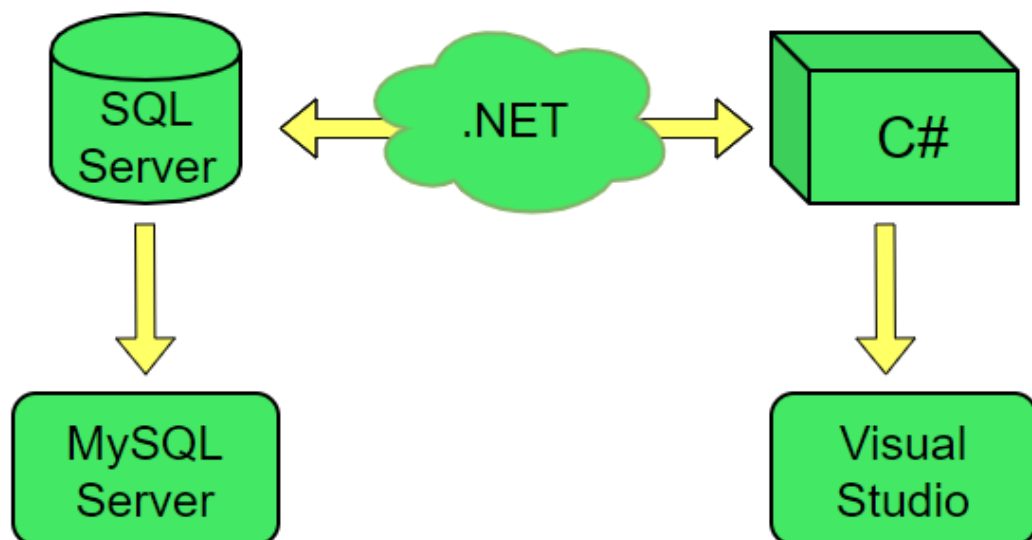


Рисунок 3.1 – Функціональний принцип роботи системи

Далі описано загальні дані про ці засоби, сфери його використання, деякі

особливості та їх властивості.

### **3.1 Опис використаної операційної системи**

Для виконання роботи була використана операційна система Windows 10 сімейства Windows NT корпорації Microsoft. На сьогоднішній день – Windows найбільш поширена операційна система виробництва Microsoft. Основними причинами вибору саме цієї операційної системи є:

- стабільність роботи системи;
- зручний інтерфейс;
- режим мультизадачної роботи;
- популярність використання;

Також необхідно відмітити, що Windows 10 має точки відновлення, тому в разі збою в роботі системи можливе повне відновлення програм, документів та інших даних, система є багатофункціональною та при встановленні має стандартний пакет драйверів [6].

### **3.2 Мова програмування C#**

Мова програмування C# – об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET, являється простою і в той же час потужною мовою програмування, що дозволяє програмістам створювати багатофункціональні програми. Розроблена Андерсом Гейлсбергом, Скотом Вілтамутом та Пітером Гольде під егідою Microsoft Research [7].

Синтаксис мови програмування C# близький до C++ і Java. Мова має сувору статичну типізацію, підтримує поліморфізм, перевантаження операторів, вказівники на функції-члени класів, атрибути, події, властивості, винятки, коментарі у форматі XML[8]. Перейнявши багато що від своїх попередників – мов C++, Delphi, Модула і Smalltalk – C#, спираючись на практику їхнього використання, виключає деякі моделі,

що зарекомендували себе як проблематичні при розробці програмних систем, наприклад, множинне спадкування класів (на відміну від C++).

Система Common Language Runtime (CLR) – «загальне середовище виконання мов» – це компонент пакету Microsoft .NET Framework, віртуальна машина, на якій виконуються всі мови платформи .NET Framework [9].

CLR транслюється початковий код в байт-код на мові IL, реалізація компіляції якого компанією Microsoft називається MSIL, а також надає MSIL-програмам (а отже, і програмам, написаним на мовах високого рівня, що підтримують .NET Framework) доступ до бібліотеки класів .NET Framework, або так званою .NET Framework Class Library (FCL).

Мова C# розроблялась спеціально як мова програмування прикладного рівня для CLR і тому вона залежить, перш за все, від можливостей самої CLR. Це стосується, перш за все, системи типів C#. Присутність або відсутність тих або інших виразних особливостей мови диктується тим, чи може конкретна мовна особливість бути трансльована у відповідні конструкції CLR. Так, з розвитком CLR від версії 1.1 до 2.0 значно збагатився і сам C#; подібної взаємодії слід чекати і надалі. (Проте ця закономірність буде порушена з виходом C# 3.0, що є розширеннями мови, що не спираються на розширення платформи .NET.) CLR надає C#, як і всім іншим .NET-орієнтованим мовам, багато можливостей, яких позбавлені «класичні» мови програмування. Наприклад, збірка сміття не реалізована в самому C#, а проводиться CLR для програм, написаних на C# точно так, як і це робиться для програм на VB.NET, J# тощо [10].

До основних переваг мови C Sharp можна віднести:

- C Sharp – це об'єктно-орієнтована мова програмування, яка користується великим попитом. Нею володіють близько 60 % усіх програмістів;
- повний і добре визначений набір основних типів;
- вбудована підтримка автоматичної генерації XML — документації;
- автоматичне звільнення динамічно розподіленої пам'яті;

- повний доступ до бібліотеки базових класів .NET, а також легкий доступ до Windows API;
- просте зміна ключів компіляції. Дозволяє отримувати виконувані файли або бібліотеки компонентів .NET, які можуть бути викликані іншим кодом так само, як елементи управління ActiveX (компоненти COM);
- має 100 000 оптимізованих методів. при чому в мові Сі Шарп є функції Делфі, Ассемблера, Сі ++ та інших мов.

### **3.3 Середовище розробки Microsoft Visual Studio 2017**

Microsoft Visual Studio – серія продуктів фірми Майкрософт, які включають інтегроване середовище розробки програмного забезпечення та ряд інших інструментальних засобів. Ці продукти дозволяють розробляти як консольні програми, так і програми з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-застосунки, як в рідному, так і в керованому кодах для всіх платформ, що підтримуються Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET [11].

Visual Studio включає в себе редактор вихідного коду з підтримкою технології IntelliSense і можливістю найпростішого рефакторингу коду. Вбудовані інструменти включають в себе редактор форм для спрощення створення графічного інтерфейсу додатку , веб-редактор, дизайнер класів і дизайнер схеми бази даних. Visual Studio дозволяє створювати і підключати сторонні доповнення (плагіни) для розширення функціональності практично на кожному рівні.

Представлення 2 серпня 2017 року Visual Studio 2017 включає .NET Framework 4.5. Головні нововведення це підтримка Windows RunTime, C++/CX(Component Extensions), бібліотека C++ AMP для GPGPU програмування, компілятор Visual C++ майже підтримує стандарт C++11. З'явився новий тип проектів, котрі дозволяють писати рідні застосунки (у стилі Windows Metro) для операційної системи Windows 8.

Основними перевагами Visual Studio є [12]:

- використання обчислювальних потужностей локального комп'ютера і хмари;
- проста реалізація спільних завдань та індивідуальний підхід;
- можливість реалізації ідей і рішень для широкого спектру платформ, включаючи Windows, Windows Server, веб-середовище, Office і SharePoint.

### 3.4 Фреймворк Entity Framework

Фреймворк ADO.NET Entity Framework (EF) - об'єктно-орієнтована технологія доступу для даних існує об'єктно-реляційне відображення (ORM) для Microsoft .NET Framework. Забезпечує можливість взаємодії з об'єктами як з LINQ у вигляді LINQ до Entities, так і з використанням Entity SQL.

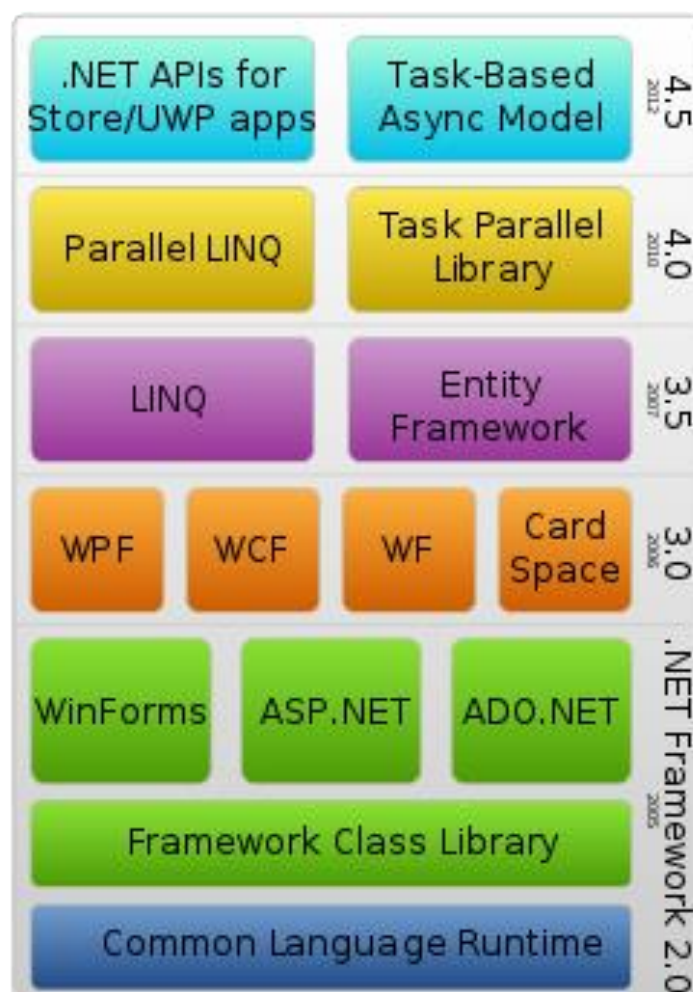


Рисунок 3.2 – Компоненти .NET Framework

Кожна реалізація .NET включає в себе компоненти, зображені на рисунку 3.2:

- одне або декілька виконавчих середовищ (приклади CLR для .NET Framework, CoreCLR і CoreRT для .NET Core);
- бібліотека класів, яка реалізує .NET Standard, а також може реалізовувати додаткові API-інтерфейси (приклади: бібліотека базових класів .NET Framework, бібліотека базових класів .NET Core);
- одна платформа додатків або кілька (приклади ASP.NET, Windows Forms і Windows Presentation Foundation (WPF) входять в .NET Framework);

Центральною концепцією Entity Framework є концепція об'єкта або суб'єкта. Сутність являє собою набір даних, пов'язаних з певним об'єктом. Тому ця технологія передбачає роботу не з таблицями, а з об'єктами та їх множинами. Будь-яка сутність, як і будь-який об'єкт з реального світу, має ряд властивостей. Властивості не обов'язково являють собою прості дані типу `int`, але можуть також представляти більш складні структури даних. І кожна сутність може мати одне або кілька властивостей, які будуть відрізняти цю сутність від інших і однозначно ідентифікувати цю сутність. Подібні властивості називаються ключами. У той же час сутність може бути пов'язана асоціативною комунікацією один-до-багатьох, один-до-одного та багатьох-до-багатьох, так само як у реальній базі даних існує з'єднання через зовнішні ключі. Відмінною особливістю Entity Framework є використання запитів LINQ для вибірки даних бази даних. За допомогою LINQ ми можемо не тільки отримувати певні лінії, які зберігають об'єкти з BD, але й отримувати об'єкти, пов'язані з різними асоціативними посиланнями. Іншою ключовою концепцією є модель даних суб'єкта. Ця модель порівнює класи об'єктів з реальними таблицями в базі даних. Модель даних об'єкту складається з трьох рівнів: концептуального, рівень сховища і рівень зіставлення (мапінга), що зображено на рисунку 3.3.

На концептуальному рівні відбувається визначення класів сутностей, які використовуються в додатку.

Рівень зіставлення (мапінга) служить посередником між попередніми двома,

визначаючи зіставлення між властивостями класу суті і стовпцями таблиць. Таким чином, ми можемо через класи, визначені у додатку, взаємодіяти з таблицями з бази даних [13].

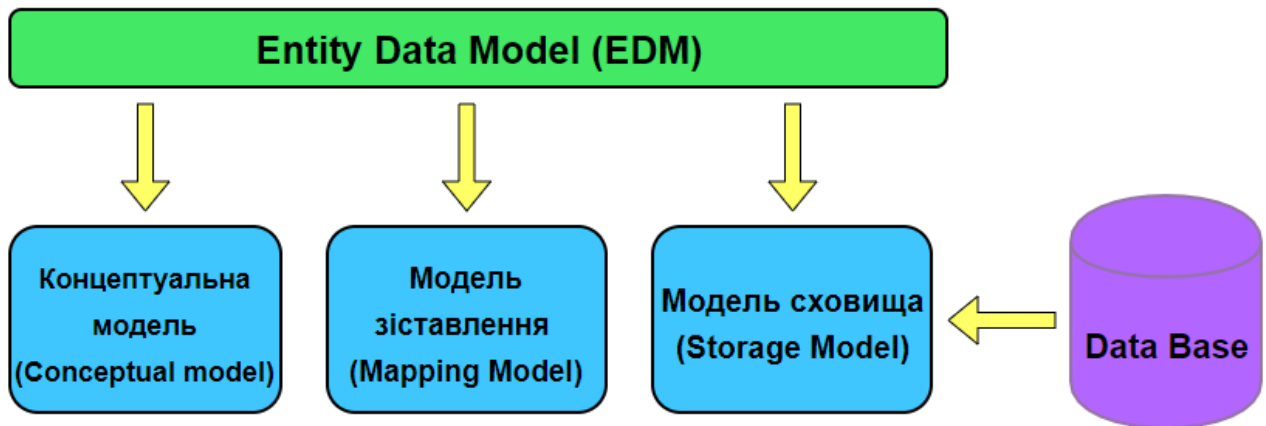


Рисунок 3.3 – Модель Entity Data Model

Фреймворк Entity Framework передбачає три можливі способи взаємодії з базою даних [14]:

— метод Database first: підходить для проектувальників баз даних - спочатку ви створюєте базу даних за допомогою різних інструментів (наприклад, SQL Server Management Studio), а потім генеруєте EDMX-модель бази даних. В даному випадку вам потрібно працювати з SQL Server і добре знати синтаксис T-SQL, але при цьому не потрібно розбиратися в C #;

— метод Model first: підходить для програмних архітекторів - спочатку ви створюєте графічну модель EDMX в Visual Studio (в фоновому режимі створюються класи C # моделі), а потім генеруєте на основі діаграми EDMX базу даних. При цьому підході не потрібно знати ні деталей T-SQL ні синтаксису C#.

— метод Code first: підходить для програмістів - при даному підході модель EDMX взагалі не використовується і ви вручну налаштовуєте класи C# об'єктної моделі (даний підхід підтримує як генерацію сутнісних класів з існуючої бази даних, так і створення бази даних зі створеної вручну моделі об'єктів C #) [14]



### **3.5 Система керування базами даних MySQL Server**

MySQL Server - система керування базами даних.

Розробку та супровід найпопулярнішої SQL-бази даних з відкритим кодом, здійснює компанія MySQL AB.

База даних являє собою структуровану сукупність даних. Ці дані можуть бути будь-якими. Для запису, вибірки і обробки даних, що зберігаються в комп'ютері, необхідна система управління базою даних, якою і є ПЗ MySQL.

Оскільки обробка великих обсягів даних є важливим для комп'ютера процесом, то управління базами даних відіграє в ньому головну роль. У реляційній базі даних MySQL дані зберігаються в окремих таблицях, це забезпечує швидкість і гнучкість. Таблиці зв'язуються між собою за допомогою відносин, це дозволяє об'єднувати дані з декількох таблиць [15].

Система керування даними MySQL Server використовується для роботи з базами даних розміром від персональних до великих баз даних масштабу підприємства або і держави.

## 4. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

Створення програмного продукту відбувається за дотримуванням певних правил, які встановлюються при обрані однієї з моделей розробки програмного забезпечення. Для реалізації поставленого завдання була обрана каскадна модель (водоспадна модель). Під моделлю зазвичай мається на увазі структура, яка визначає послідовність виконання та взаємозв'язку процесів, дій і завдань протягом життєвого циклу. Схема каскадного процесу розробки програмного забезпечення зображена на рисунку 4.1.

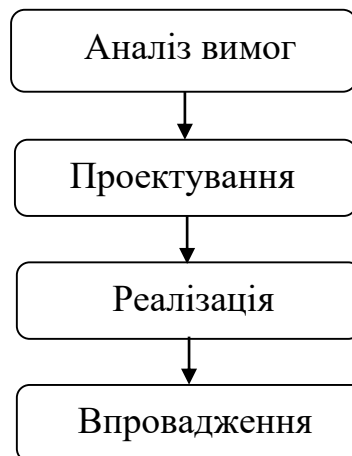


Рисунок 4.1 – Каскадна модель розробки програмного забезпечення

В каскадній моделі, розробник переходить від однієї стадії до іншої послідовно. Спочатку повністю завершується етап «визначення вимог», в результаті чого виходить список вимог до ПЗ. Після того як вимоги повністю визначені, відбувається перехід до проектування, в ході якого створюються документи, що детально описують для програмістів спосіб і план реалізації зазначених вимог. Після того як проектування повністю виконане, програмістами виконується реалізація отриманого проекту. На наступній стадії процесу відбувається інтеграція окремих компонентів, що розробляються різними командами програмістів. Після того як реалізація та інтеграція завершені, проводиться тестування і налагодження продукту; на цій стадії усуваються всі недоліки, що з'явилися на попередніх стадіях розробки. Після цього

програмний продукт впроваджується і забезпечується його підтримка – внесення нової функціональності та усунення помилок.

Тим самим, каскадна модель передбачає, що перехід від однієї фази розробки до іншої відбувається тільки після повного і успішного завершення попередньої фази, і що переходів назад або вперед або перекриття фаз – не відбувається.

Реалізація будь-якої системи потребує перш за все аналізу та пошуку оптимального шляху вирішення цього завдання. Після аналізу постановки задачі було виділено основні критерії, яким має задовольняти система з точки зору програмної реалізації:

1) незалежна розробка різних частин програмного продукту для забезпечення швидкого нарощування функціоналу – з цього виникає необхідність відокремлення різних частин системи одна від одної;

2) архітектурна можливість розширення наявних можливостей системи без зміни інтерфейсу її роботи з користувачем.

Після аналізу цих вимог було вирішено, що найбільш повно цим критеріям задовольняє модульний принцип побудови архітектури системи з використанням об'єктно-орієнтованого програмування, за допомогою якого можна досить легко вирішити проблеми з майбутнім розширенням системи. Модульний принцип організації також має наступні переваги:

1) розподілена розробка системи сприяє швидшому пошуку помилок на відміну від програм, де система взаємодіє як монолітний комплекс без можливості вільної зміни окремого її елемента, що призводить до низької масштабованості таких систем;

2) кожен модуль здатен надавати певний набір методів та можливостей для інших модулів – так званий інтерфейс модуля, який робить можливим стандартизацію протоколів міжмодульної взаємодії;

3) інтерфейс модуля з точки зору проектування є незмінним елементом (і може бути змінений лише при перепроєктуванні системи), проте спосіб реалізації цього інтерфейсу може змінюватися. Таким чином досягається легкість тестування нових можливостей модуля, шляхом поступової зміни способу реалізації

інтерфейсу;

4) для великих проектів, що знаходяться у постійній розробці, важливим є час компіляції. За умови незмінності інтерфейсу модуля є можливість приєднувати до основної програми різні реалізації одного й того самого модуля, відкомпільовані незалежно від основної програми. Таким чином, час компіляції всього проекту зменшується, що робить процес розробки та тестування швидшим.

Реалізація модульної системи потребує, перш за все, правильного розподілу виконуваних завдань між модулями, а також оптимальності взаємодії між ними, що досягається попереднім аналізом та проектуванням. Поставлена задача в самій своїй суті потребує модульного вирішення, оскільки повинна забезпечувати функціональність на рівні бібліотеки класів для подальшого використання. Таким чином, перш ніж конкретизувати внутрішній склад кожного модуля, необхідно визначити та реалізувати той рівень абстракції, що буде відображати систему на рівні, достатньому для аналізу внутрішніх елементів кожного модуля, не відволікаючись на інші. А це і є основний принцип модульного програмування.

Крім структури модулів системи необхідно також розглянути технології, що будуть використані при розробці програмного продукту.

Перш за все – це техніка об'єктно-орієнтованого програмування. Вона полягає в komponуванні програми як ієрархічної системи класів, що відображають собою поняття предметної області. Кожен клас являє собою сукупність даних, що характеризують його, та методів, якими він взаємодіє з іншими класами. Методи класу – його невід'ємні елементи, що відображають собою можливості класу, доступні для застосування в програмі. В термінології фреймових моделей методи – це приєднані процедури або демони (в залежності від функції, що вони виконують в класі) певного фрейму(класу). Об'єктно-орієнтоване програмування (в подальшому ООП) ґрунтується на трьох основних принципах:

1) Інкапсуляція (encapsulation) – це механізм, що поєднує дані та програмний код, який маніпулює цими даними, а також захищає їх від зовнішнього втручання або неправильного використання. Результатом поєднання коду та даних є конкретний об'єкт, який з точки зору сучасних мов програмування є екземпляром

певного класу. За концепцією ООП всередині класу методи та дані можуть бути закритими (private). Закриті частини класу доступні лише для інших частин цього ж об'єкту і недоступні для всіх інших, які існують поза об'єктом. Тривіальною є ситуація, коли відкрита частина методів класу забезпечує контрольований доступ до закритих даних, забезпечуючи їх безпечне редагування без ризику змін, що будуть фатальними для стану об'єкта.

2) Поліморфізм (polymorphism) – це властивість, яка дозволяє створити єдиний інтерфейс для набору схожих дій, що дозволяє структурно поєднати їх. В загальному випадку це означає створення єдиного інтерфейсу, якому задовольняють усі поліморфні елементи. Поліморфізм дозволяє змінювати поведінку класів в межах інтерфейсу, що, в свою чергу, дозволяє створювати класи, які матимуть однакове загальне призначення, проте будуть відрізнятися його реалізацією. Важливе значення для поєднання класів та інтерфейсів має механізм наслідування.

3) Наслідування (inheritance) – це процес, за допомогою якого один об'єкт може набувати властивостей іншого. Точніше, об'єкт може наслідувати властивості іншого об'єкта та додавати до них риси, характерні лише для нього. Концепція наслідування дозволяє підтримувати концепцію ієрархії класів, що робить керованою та прогнозованою поведінку великих потоків інформації. За допомогою наслідування можна виділити спільні характеристики декількох класів в спільний для них клас-предок, від якого вони будуть наслідувати спільну поведінку та при необхідності коригувати її так як це необхідно. Важливо також виділити поняття інтерфейсу – повністю абстрактного класу, який задає лише спосіб взаємодії з собою не реалізуючи її, залишаючи реалізацію класам-нащадкам.

Через поєднання цих трьох принципів досягається вся гнучкість та сила об'єктно-орієнтованого програмування. Таким чином, проектування програми являє собою процес відображення елементів предметної області в сукупність ієрархічних об'єктів, що взаємодіють між собою так само, як це відбувається в самій предметній області. Цей метод не розриває сприйняття понять від їхньої реалізації, тому з допомогою нього можна досягти набагато більшого контролю над поведінкою програми ніж в класичному процедурному програмуванні. Вибір такого методу для

реалізації програмного продукту продиктований, у першу чергу, необхідністю створити надійний та простий у використанні продукт, поведінка якого буде легко прогнозованою при інтеграції його з іншими програмами.

У якості мови програмування для реалізації системи було взято мову C#, що має підтримку усіх необхідних технік програмування, є універсальною та високорівневою мовою програмування, придатною для створення програм будь-якої складності

## **4.1 Архітектура програмного продукту**

Архітектура програмного продукту (software architecture) - це структура програми або обчислювальної системи, яка включає програмні компоненти, видимі зовні властивості цих компонентів, а також відносини між ними. Цей термін стосується також документування архітектури програмного забезпечення. Документування архітектури програмного продукту спрощує процес комунікації між зацікавленими особами, дозволяє зафіксувати прийняті на ранніх етапах проектування рішення про високорівневий дизайн системи і дозволяє використовувати компоненти цього дизайну і шаблони повторно в інших проектах.

Прикладне програмне забезпечення, призначено для розв'язання задач користувача. Виконання роботи розподілене на окремі блоки, які можуть виконуватись як послідовно, так і паралельно (в окремих потоках програмного продукту). Кожний блок виконує тільки свою специфічну роботу, наприклад перевіряє дані введені користувачем. На рисунку 4.2 зображено взаємодію блоків програмного продукту, які складають архітектуру програмного продукту.

Формальні методи опису програм ґрунтуються на специфікаціях, аксіомах, описах деяких попередніх умов, твердженнях і постулатах, що визначають заключну умову одержання програмою правильного результату. Специфікація функцій і даних, якими ці функції оперують, а також умови і твердження – основа доведення правильності програми.

В архітектурі застосунку можна виділити два головних компонента:

- модуль графічного інтерфейсу користувача, який формує зовнішній

вигляд програмного продукту та виконує взаємодію з користувачем програмного продукту;

— модуль розв’язку задачі, який отримує та аналізує вхідні дані, складає та розв’язує задачу, оформлює результат виконання та передає його до графічного інтерфейсу.



Рисунок 4.2 – Взаємодія структурних блоків програмного продукту

Графічний інтерфейс користувача дозволяє у зручному для користувача форматі взаємодіяти з програмним продуктом. Він складається з двох під-блоків:

- блок роботи з вхідними даними;
- блок роботи з вихідними даними, який будує графічне зображення процесу.

Модуль розв’язку задачі виконує найголовнішу роботу програмного продукту.

Складається модуль з трьох компонентів:

- блок обробки результату моделювання групує результат виконання блоку розв’язку задачі у набір масивів даних та передає до блоку графічного інтерфейсу користувача;

— блок розв'язку задачі розрахунку нормованих показників та показника сумарного питомого нормованого потенціалу.

Оскільки програмний продукт написаний на мові, яка реалізує принципи об'єктно-орієнтованого програмування, кожний з компонентів програми можна представити у вигляді об'єкта, яких є екземпляром певного класу. Тобто робота та поведінка кожного компонента архітектури описується класом. Це дає змогу створити ефективну та гнучку взаємодію об'єктів. Безперечною перевагою використання об'єктно-орієнтованого підходу є концептуальна близькість до предметної області довільної структури та призначення. Механізм спадкоємства атрибутів і методів дозволяє будувати похідні поняття на основі базових і таким чином створювати модель складної предметної області з заданими властивостями.

Для покращення швидкодії програмного продукту, всі трудомісткі операції виконуються у окремих паралельних потоках. Такий підхід скорочує час виконання операцій в рази на комп'ютерах з багатоядерними процесорами, оскільки кожен з потоків по можливості буде виконуватись на окремому ядрі. Відповідальність за керування та синхронізацію робочих потоків бере на себе виконуюче середовище Common Language Runtime (CLR) платформи .Net.

В основу синхронізації покладено поняття блокування, за допомогою якої організовується управління доступом до кодового блоку в об'єкті. Коли об'єкт заблокований одним потоком, інші потоки не можуть отримати доступ до заблокованого кодового блоку. Коли ж блокування знімається одним потоком, об'єкт стає доступним для використання в іншому потоці.

## **4.2 Структура проекту**

Структура проекту складається з наступних модулів (рисунок 4.3):

- модуль Region;
- модуль DAO (Data Access Object) – встановлює доступ і зв'язок до бази даних;



- модуль Form1;
- модуль MapForm;
- файл App.config.

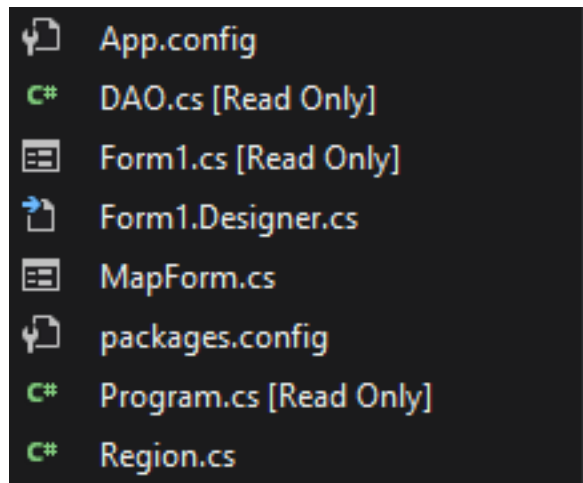


Рисунок 4.3 – Структура модулів проекту

Детальніше розглянемо кожен із модулів проекту.

### **Модуль Data Access Object**

З'єднання з базою даних є однією з найважливіших складових програми. Завжди виділяється частина коду або модуль, що відповідає за встановлення доступу до бази даних, передачу запитів в бд і обробку отриманих від неї відповідей. DAO абстрагує сутності системи і робить їх відображення на БД, визначає загальні методи використання з'єднання, його отримання, закриття та (або) повернення в Connection Pool.

Вершиною ієрархії DAO є абстрактний клас або інтерфейс з описом загальних методів, які використовуються при взаємодії з базою даних. Як правило, це методи пошуку, внесення, оновлення, видалення даних та інші.

### **Модуль Region**

Модуль Region являє собою набір класів на мові програмування C#, згенерований автоматично Entity Framework при встановленні зв'язку з базою даних.

У цьому модулі зберігається інформація про таблиці бази даних та їх поля, методи оновлення бази даних.

### **Модуль Form1**

У модулі Form1 проводяться основні обрахунки, а саме обчислюються нормалізовані показники всіх ВДЕ та обчислюється сумарний питомий нормований потенціал для кожного регіону.

Також у цьому модулі містяться:

- методи збереження обрахованих показників у відповідну таблицю бази даних;
- методи графічного відображення цієї таблиці на головному користувацькому вікні програми з використанням бібліотеки DataGridView;
- методи роботи кнопок, що містяться на головному користувацькому вікні.

### **Модуль MapForm**

У модулі MapForm реалізовується графічна інтерпретація вихідних даних у вигляді стовпчастих діаграм для кожного регіону України за допомогою методів DrawDiagram бібліотеки Drawing.

Мапа України поділена на адміністративні регіони – слугує фоном на координатній площині, для виведення стовпчастої діаграми для певного регіону, необхідно визначити координати точки на площині, що належить цьому регіону і передати їх у метод DrawDiagram.

### **Файл App.config**

App.config це XML файл, у якому міститься інформація про додаток, модулі, що використовуються, рядок підключення до бази даних та ключі для доступу до API проекту.

## **4.3 Опис бази даних проекту**

База даних являє собою набір зв'язаних між собою таблиць. Концептуальна схема бази даних зображена на рисунку 4.4.

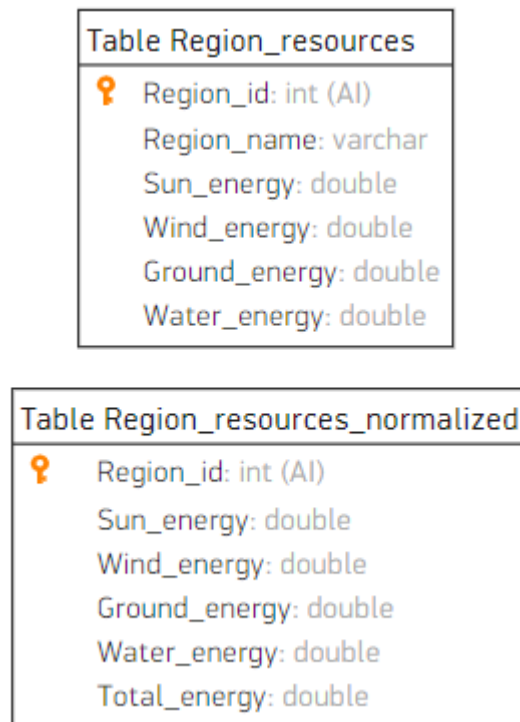


Рисунок 4.4 – Концептуальна схема бази даних

БД складається з 2 таблиць:

- таблиці Region resources;
- таблиці Region resources normalized.

В таблиці “Region resources”, зберігаються показники ресурсів ВДЕ для кожного регіону: Sun\_energy, Wind\_energy , Ground\_energy , Water\_energy, саме вони і є вхідними даними в програмі (рисунок 4.5).

Поле Region\_id унікально ідентифікує таблицю і є головним ключем.

	№	Область	Показник сонячної радіації	Показник середньої швидкості вітру	Показник рівня енергії землі	Показник гідроенергії
►	1	Crimea	0,38	4,70	1,11	0,05
	3	Volynska	0,18	0,20	0,24	0,03
	4	Dnipropetrovska	0,32	0,70	0,38	0,02
	5	Donetska	0,27	2,27	0,32	0,05
	6	Zhutomyska	0,26	0,30	0,36	0,08
	7	Zakarpatska	0,14	0,73	0,85	1,11
	8	Zaporizhska	0,28	4,10	0,36	0,01
	9	Ivano-Frankivska	0,13	0,27	0,18	0,10
	10	Kyivska	0,26	0,28	0,35	0,05

Рисунок 4.5 – Фрагмент таблиці “Region resources”

В таблиці “Region resources normalized”, зберігаються нормалізовані показники ресурсів ВДЕ: Sun\_energy, Wind\_energy, Ground\_energy, Water\_energy, та показник сумарного нормованого потенціалу Total\_energy для кожного регіону. Ці дані є вихідними в програмі, на їх основі користувач проводить оцінку ресурсів відновлюваних джерел енергії території (рисунок 4.6).

	№	Область	Питомий нормований потенціал сонячної енергії	Питомий нормований потенціал вітрової енергії	Питомий нормований потенціал енергії землі	Питомий нормований потенціал гідроенергії	Сумарний питомий нормований потенціал всіх видів енергії
►	1	Crimea	1,0000	1,0000	1,0000	0,0364	3,0364
	3	Volynska	0,3103	0,0132	0,1635	0,0182	0,5051
	4	Dnipropetrovska	0,7931	0,1228	0,2981	0,0091	1,2231
	5	Donetska	0,6207	0,4671	0,2404	0,0364	1,3645
	6	Zhutomyska	0,5862	0,0351	0,2788	0,0636	0,9638
	7	Zakarpatska	0,1724	0,1294	0,7500	1,0000	2,0518

Рисунок 4.6 – Фрагмент таблиці “Region resources normalized”

## **5. МЕТОДИКА РОБОТИ КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ**

В даному розділі описується методика роботи користувача з програмним продуктом.

Для роботи системи необхідно запустити програму на комп'ютері. Користувач потрапляє на стартове вікно, з якого він може здійснити дії щодо ознайомлення з нормалізованими показниками ВДЕ для кожного регіону України. Дані представлені у вигляді таблиці та у вигляді діаграм.

### **5.1 Інсталяція та системні вимоги**

Для роботи користувача з розробленою програмною системою користувачу потрібні мінімальні потужності апаратного забезпечення:

- IBM-сумісний комп'ютер з процесором класу Pentium-III з тактовою частотою 200 MHz;
- об'єм оперативної пам'яті (RAM) – 128МБ;
- операційна система Microsoft Windows 7/8/10;
- 10МБ дискового простору;
- монітор SVGA (800 x 600), клавіатура.

Для встановлення системи потрібна наявність наступних системних засобів:

- 1) додаток Microsoft Visual Studio 2017/2018/2019;
- 2) сервер БД MySQL Server;
- 3) додаток MySQL Workbench.

Для інсталяції програми необхідно відкрити файл UAEnergy.exe.

## 5.2 Навігація програмної системи

Користувач системи має можливість користуватися простим і зручним інтерфейсом, основні елементи якого інтуїтивно зрозумілі. Після запуску програми з'являється головна форма (рисунок 5.1).



	№	Область	Показник сонячної радіації	Показник середньої швидкості вітру	Показник рівня енергії землі	Показник гідроенергії
▶	1	Crimea	0,38	4,70	1,11	0,05
	3	Volynska	0,18	0,20	0,24	0,03
	4	Dnipropetrovska	0,32	0,70	0,38	0,02
	5	Donetska	0,27	2,27	0,32	0,05
	6	Zhutomyska	0,26	0,30	0,36	0,08
	7	Zakarpatska	0,14	0,73	0,85	1,11
	8	Zaporizhska	0,28	4,10	0,36	0,01
	9	Ivano-Frankivska	0,13	0,27	0,18	0,10
	10	Kyivska	0,26	0,28	0,35	0,05
	11	Kharkivska	0,22	0,50	0,20	0,04

Обчислити нормовані показники

Мапа України

Рисунок 5.1 – Головна форма програми

Головна форма носить ознайомлювальний характер та інформує користувача про кількість ресурсів відновлюваних джерел енергії різних типів у всіх регіонах України. Ці дані представлені у вигляді таблиці з можливістю сортування за різними критеріями.

Щоб ознайомитись з нормалізованими показниками ресурсів ВДЕ потрібно натиснути кнопку “Обчислити нормовані показники” (рисунок 5.2).

Після цього, з'явиться нова таблиця з нормалізованими показниками та з обрахованим показником сумарного нормованого питомого потенціалу для кожного регіону України. Дані також представлені у вигляді таблиці.

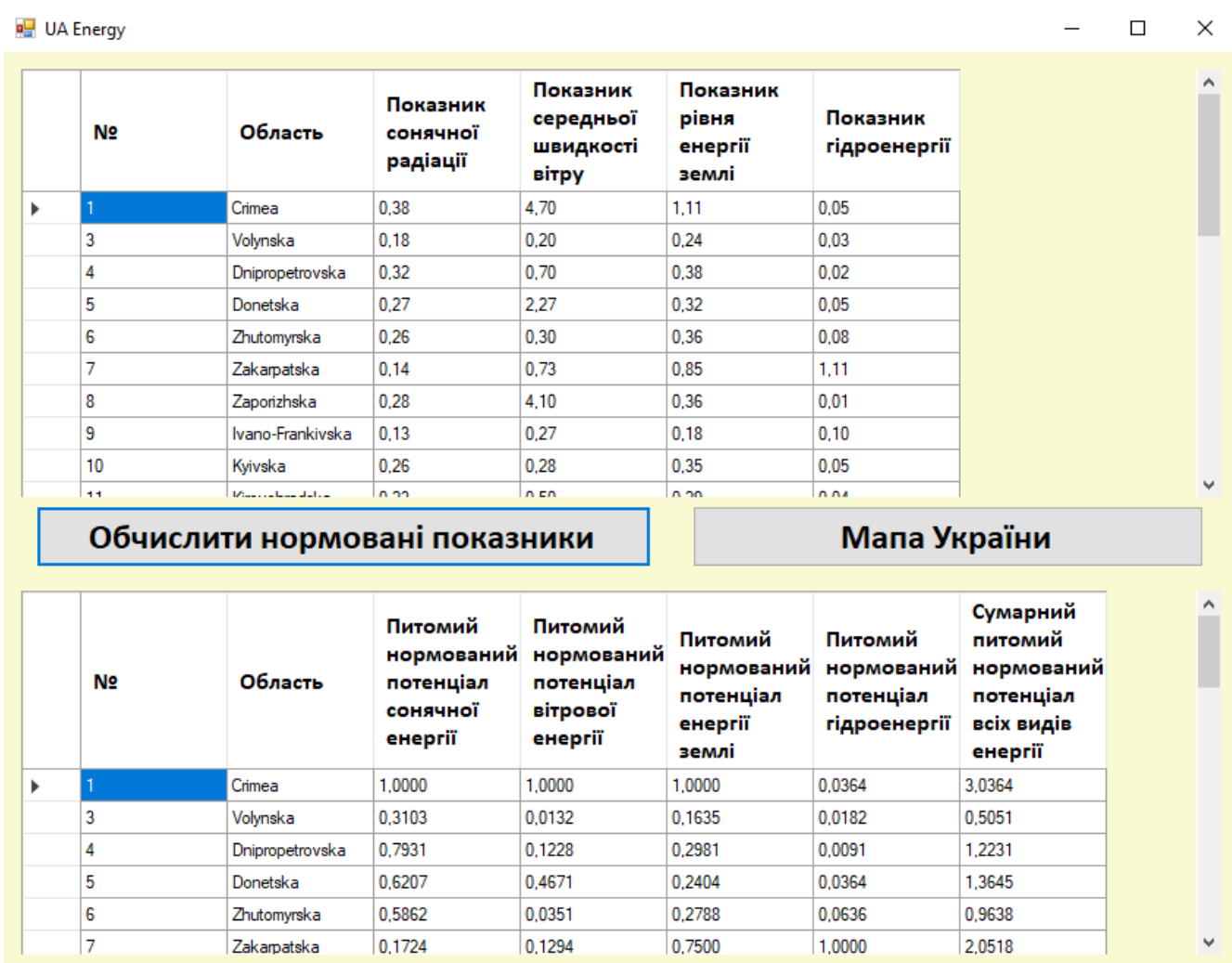


Рисунок 5.2 — Обчислення нормованих показників ВДЕ

Для проведення якісної оцінки потенціалу забезпеченості України ресурсами ВДЕ необхідно натиснути кнопку “Мапа України”.

У новому вікні буде зображено карту України, поділену на адміністративні регіони (рисунок 5.3). На кожній області зображено стовпчасту діаграму з нормованими показниками:

- сонячної енергії ;
- енергії води;
- вітрової енергії;
- геотермальної енергії;
- сумарного питомого нормованого потенціалу всієї енергії для регіону.



Рисунок 5.3 – Мапа України з показниками ВДЕ для кожного регіону



## 6. Аналіз результатів роботи програмної системи

На основі результатів роботи програмної системи графічно інтерпретованих у вигляді діаграм дуже зручно провести порівняння показників ресурсів ВДЕ, оцінити потенціал відновлюваних джерел енергії регіонів та провести ресурсне районування території України за цими показниками.

### 6.1 Районування території за ресурсним потенціалом вітрової енергії

В результаті проведення нормування статистичних метеорологічних показників по середній швидкості вітру на висоті 10 метрів, пропонується районування території України за ресурсним потенціалом вітрової енергії, що зображене на рисунку 6.1.

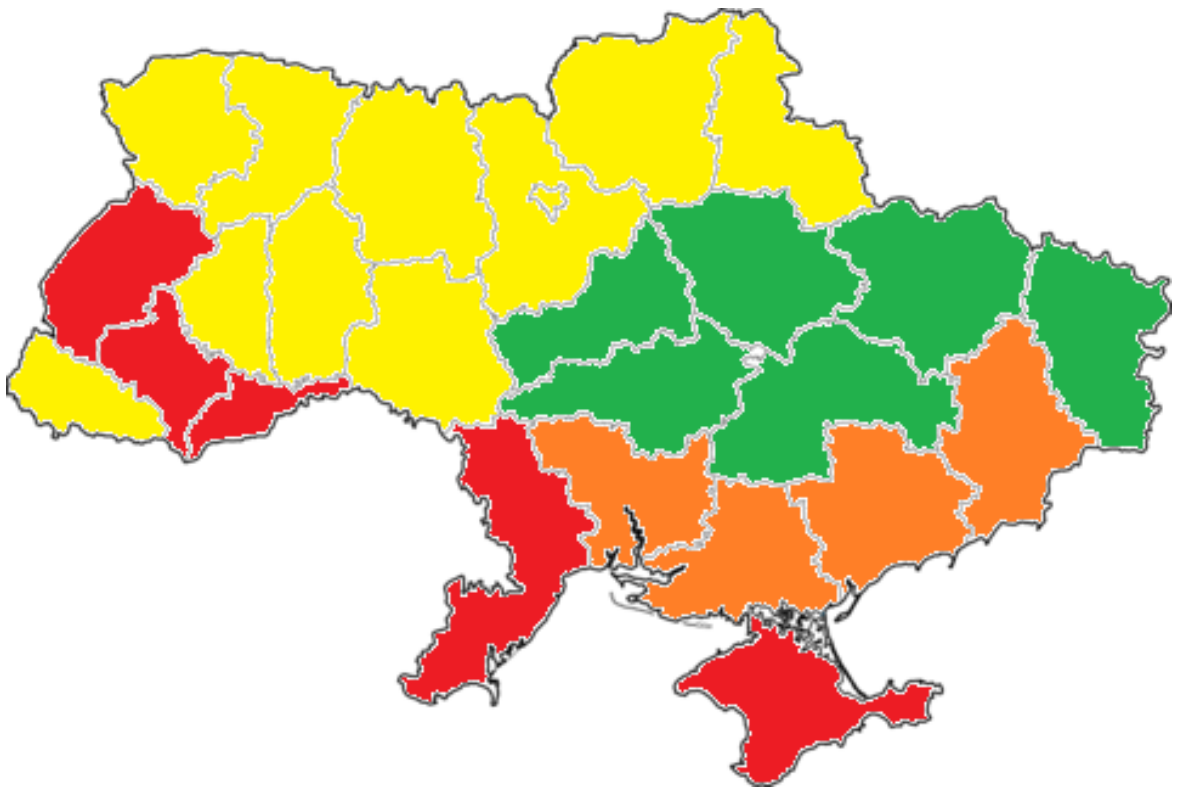


Рисунок 6.1 – Районування території України за ресурсним потенціалом вітрової енергії

Територія, зображена жовтим кольором має незначний потенціал.

Територія, зображена зеленим кольором має помірний потенціал ресурсів вітрової енергії.

Територія, зображена оранжевим кольором – має значний потенціал, а територія, зображена червоним – має найвищий потенціал вітрової енергетики України.

## **6.2 Районування території за ресурсним потенціалом сонячної енергії**

За отриманими результатами роботи програмної системи пропонується районування території України за ресурсним потенціалом сонячної енергії, що зображене на рисунку 6.2.

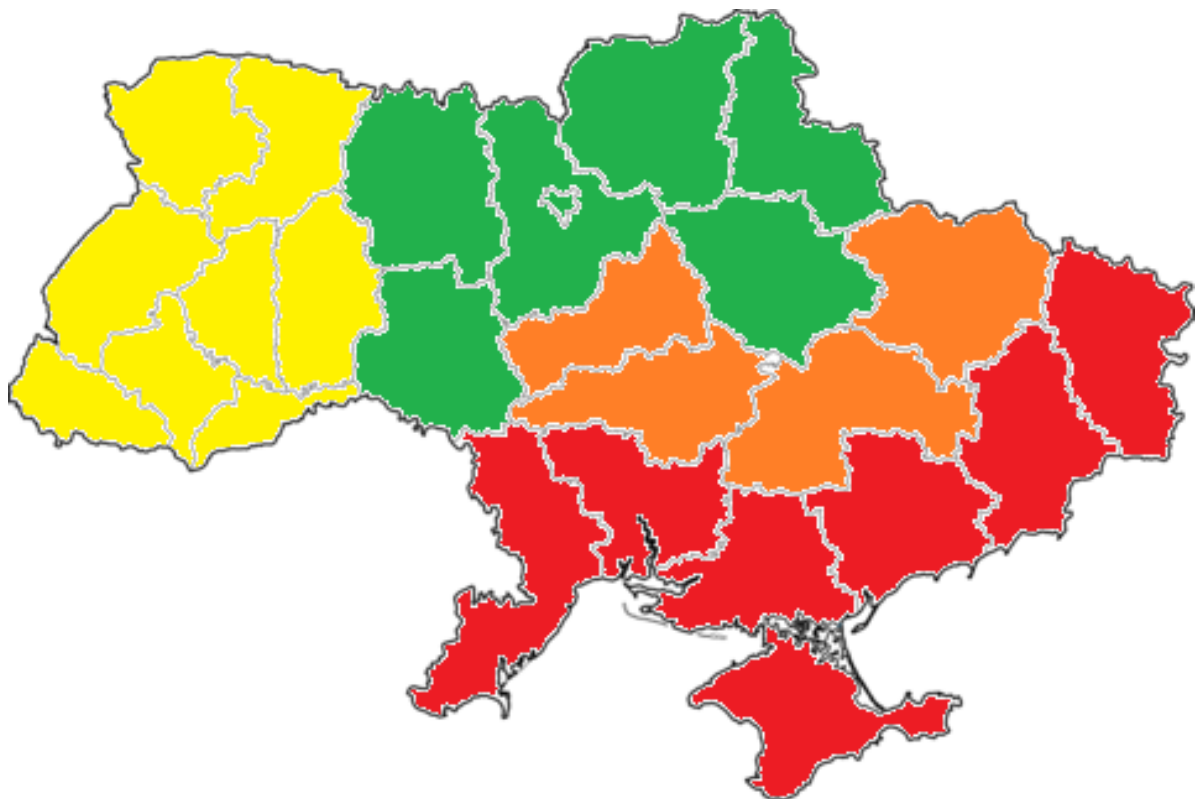


Рисунок 6.2 – Районування території України за ресурсним потенціалом сонячної енергії

Згідно з даними, зображеними на рисунку, Західна Україна має незначний

потенціал ресурсів сонячної енергії. Північна та центральна – середній потенціал, а найбільший потенціал забезпечуваності ресурсами сонячної енергії має Східна та Південна частина України.

### **6.3 Районування території України за гідроенергетичним потенціалом**

Згідно з результатами роботи програми, Україна має досить високий потенціал забезпечуваності водними ресурсами. Районування території можна провести на основі аналізу нормованих показників так, як зображено на рисунку 6.3.

Східна та центральні області мають не дуже значний гідроенергетичний потенціал, а Західна та Південна частини України навпаки – дуже добре забезпечені ресурсами енергії води.

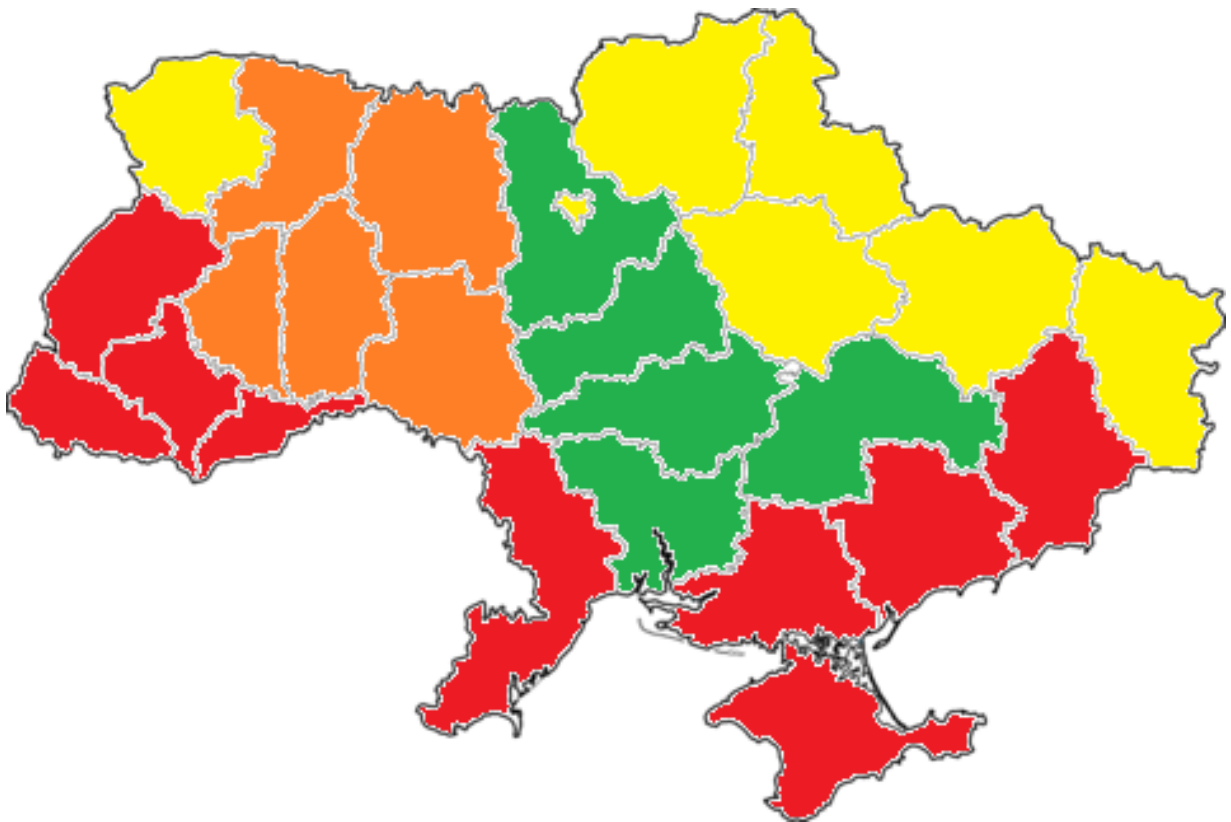


Рисунок 6.3 – Районування території України за ресурсним потенціалом гідроенергії

## 6.4 Районування території України потенціалом геотермальної енергії

На жаль, не всі області мають високий потенціал геотермальної енергії в Україні. Питомі нормовані показники деяких областей дорівнюють нулю, оскільки є зовсім незначними у порівнянні з іншими.

На основі цих показників можна провести таке районування території України за потенціалом геотермальної енергії.

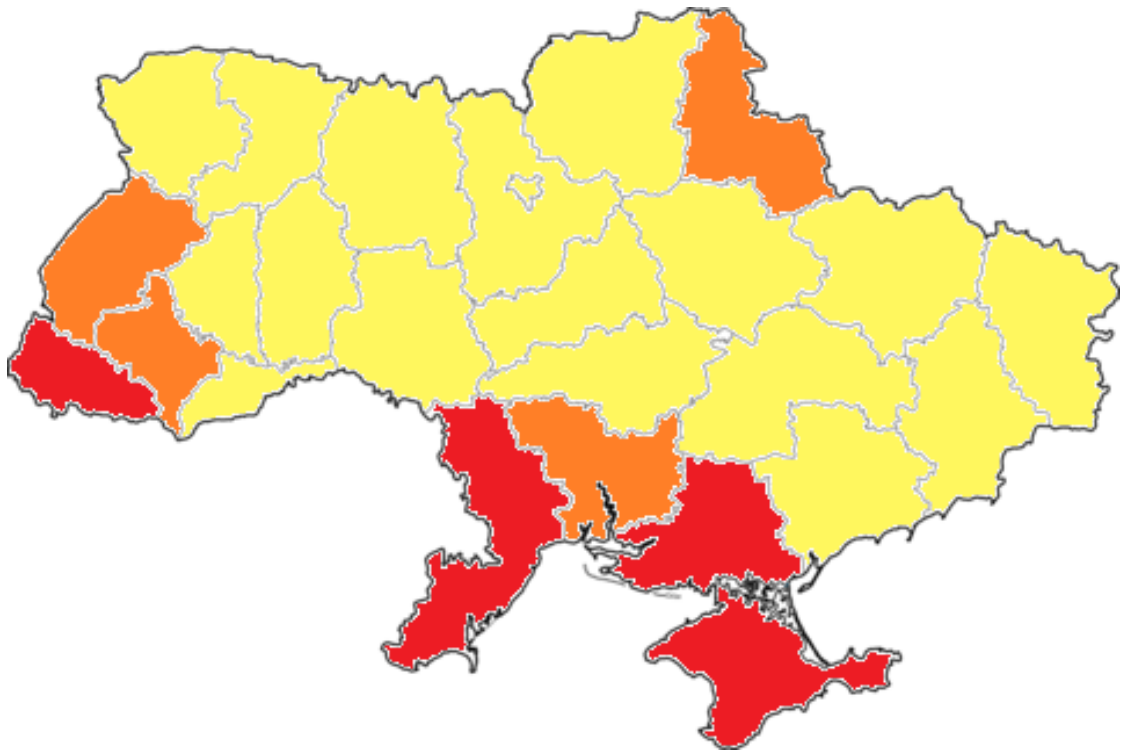


Рисунок 6.2 – Районування території України за потенціалом геотермальної енергії

# ВИСНОВКИ

Розглянута задача оцінки ресурсів відновлюваних джерел енергії території з використанням показника сумарного питомого нормованого потенціалу.

Розроблено програмну систему, яка дозволяє проводити оцінку ресурсів відновлюваних джерел енергії території використавши сучасні програмно-технічні рішення. Програмний комплекс розроблено на платформі MS Visual Studio 2017 на мові високого рівня С# з графічним інтерфейсом і можливістю виведення нормованих питомих показників відновлювальних джерел енергії, а саме: сонячної, вітрової, геотермальної та гідроенергії, на екран у вигляді таблиці та у вигляді стовпчастих діаграм на мапі України.

В результаті створення програмної системи було показано, що в Україні високий потенціал ВДЕ, що є суттєвим вкладом в енергетичну незалежність України. А також проведено районне територіальне розподілення областей України за потенціалом різних типів відновлювальних джерел енергії.

Користувачами даної програмної системи можуть бути спеціалісти міністерства енергетики, спеціалісти територіального районування, інженери з проектування енергетичних процесів та систем, фахівці підприємств з енергозабезпечення, а також зацікавлені спеціалісти інших профілів.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Севастьянов Р.В. Проблемы та перспективи енергозабезпечення на промислових підприємствах. [Електронний ресурс] / Севастьянов Р.В — Режим доступу до ресурсу: <http://eir.pstu.edu/bitstream/handle/123456789/5537/%D0%A1.%20107.pdf?sequence=1>.
2. Сдасюк Г.В. М.М. Барановський – основоположник вітчизняної економічної географії. [Електронний ресурс] — Режим доступу: <https://elibrary.ru/item.asp?id=26089160>.
3. Сайт StudFiles // Регіональна економіка. [Електронний ресурс] — Режим доступу: <https://studfiles.net/preview/5149715/>
4. Нефедова Л. В. Метод типологии территорий на основе комплексной оценки потенциала ресурсов возобновляемых источников энергии. [Електронний ресурс] / Л. В. Нефедова // Электронный научный журнал «Исследовано в России». — Режим доступу: <http://zhurnal.aep.relarn.ru/articles/2007/146.pdf2>. Вітроенергетика / за ред. Д. де Рензо / пер. з англ. Зубарева В.В. [за ред. Шефтер Я.І.]. Москва, 1982.
5. Сайт Державна служба статистики України // Energospizhyvannia alternatyvnoi enerhii [Електронний ресурс] — Режим доступу: <http://www.ukrstat.gov.ua/>
6. Сайт Microsoft // Windows 10 [Електронний ресурс] — Режим доступу: <https://www.microsoft.com/uk-ua/windows>
7. Visual C# 2008: базовый курс / [К. Уотсон, К. Нейгел, Я. Педерсен та ін.]. — М.: И.Д. Вильямс, 2009. — 1216 с.
8. Відомості про проекти і рішення на мові C# [Електронний ресурс] — Режим доступу до ресурсу: <https://docs.microsoft.com/ru-ru/visualstudio/get-started/tutorial-projects-solutions?view=vs-2019>.
9. Шарп Д. Microsoft Visual C#. Подробное руководство / Дж. Шарп. — СПб: Питер, 2017. — 848 с. — (8-е издание).
10. C# / [Х. Дейтел, П. Дейтел, Д. Листфилд та ін.]. — СПб: БХВ-Петербург, 2006. — 1056 с.

11. Сайт Microsoft // Visual Studio 2017 [Электронный ресурс] – Режим доступа до ресурсу: <https://visualstudio.microsoft.com/ru/>
12. Описание среды разработки Microsoft Visual Studio [Электронный ресурс] – Режим доступа до ресурсу: [https://studbooks.net/2258619/informatika/opisanie\\_sredy\\_razrabotki\\_microsoft\\_visual\\_studio](https://studbooks.net/2258619/informatika/opisanie_sredy_razrabotki_microsoft_visual_studio)
13. Руководство по Entity Framework [Электронный ресурс] – Режим доступа до ресурсу: <https://metanit.com/sharp/entityframework/>
14. Работа с Entity Framework 6 [Электронный ресурс] – Режим доступа до ресурсу: <https://professorweb.ru/my/entity-framework/6/level1/>
15. MySQL 8.0. Полное руководство [Электронный ресурс] – Режим доступа до ресурсу: <http://www.rldp.ru/mysql/mysql80/index.htm>

# ДОДАТОК 1

Автоматизована система оцінки ресурсів відновлюваних джерел енергії  
території з використанням показника сумарного питомого нормованого  
потенціалу

## СПЕЦИФІКАЦІЯ

УКР.КПІм.ІгоряСікорського\_ТЕФ\_АПЕПС\_ТР-5288\_19Б

Аркушів 2

Київ 2019



Позначення	Найменування	Примітки
Документація		
УКР.КПІм.ІгоряСікорського_ТЕ Ф_АПЕПС_ТР5288_19Б 81-1	Записка.docx	Текстова частина дипломної роботи
Компоненти		
УКР.КПІм.ІгоряСікорського_ТЕ Ф_АПЕПС_ТР5288_19Б 12-1	UAEnergy.exe	Основний компонент додатку
УКР.КПІм.ІгоряСікорського_ТЕ Ф_АПЕПС_ТР5288_19Б 13-1	Текст програмного модулю	
УКР.КПІм.ІгоряСікорського_ТЕ Ф_АПЕПС_ТР5288_19Б 14-1	Опис програми	

## ДОДАТОК 2

Автоматизована система оцінки ресурсів відновлюваних джерел енергії  
території з використанням показника сумарного питомого нормованого  
потенціалу

ТЕКСТ ПРОГРАМНОГО МОДУЛЮ

УКР.КПІм.ІгоряСікорського\_ТЕФ\_АПЕПС\_ТР5288\_19Б

Аркушів 6

Київ 2019

## Тест програми оцінки ресурсів відновлюваних джерел енергії території з використанням показника сумарного питомого нормованого потенціалу

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace UAEnergy
{
    public class Region
    {
        public Int32 Id { get; set; }
        public String Name { get; set; }
        public Double SunEnergyAmount { get; set; }
        public Double WindEnergyAmount { get; set; }
        public Double GroundEnergyAmount { get; set; }
        public Double WaterEnergyAmount { get; set; }

        public Double SunEnergyNormalizedAmount { get; set; }
        public Double WindEnergyNormalizedAmount { get; set; }
        public Double GroundEnergyNormalizedAmount { get; set; }
        public Double WaterEnergyNormalizedAmount { get; set; }
        public Double TotalEnergyNormalizedAmount { get; set; }

        public Region(int id, string name, double sun, double wind, double ground, double water)
        {
            Id = id;
            Name = name;
            SunEnergyAmount = sun;
            WindEnergyAmount = wind;
            GroundEnergyAmount = ground;
            WaterEnergyAmount = water;
        }
    }

    class DAO
    {
        private Random random = new Random();
        private MySqlConnection connection;
        private String connectionInfo = "Database=ua_energy;Data Source=localhost;User Id=root;Password=pass";

        public DAO()
        {
            CreateConnection();
        }

        private void CreateConnection()
        {
            connection = new MySqlConnection(connectionInfo);
        }

        public List<Region> LoadFromDB()
        {

```

```

String query1 = "select * from region_resources order by region_id";
MySQLCommand command = new MySQLCommand(query1, connection);
MySQLDataReader reader = null;
List<Region> regions = new List<Region>();
try
{
    connection.Open();
    reader = command.ExecuteReader();
    while (reader.Read())
    {
        Region region = new Region(Int32.Parse(reader[0].ToString()),
            reader[1].ToString(),
            Double.Parse(reader[2].ToString()),
            Double.Parse(reader[3].ToString()),
            Double.Parse(reader[4].ToString()),
            Double.Parse(reader[5].ToString()));
        regions.Add(region);
    }
    reader.Close();
    connection.Close();

    return regions;
}
catch (Exception e)
{
    System.Windows.Forms.MessageBox.Show(e.Message + "\r\nПомилка під час завантаження даних з бази");
    return null;
}
finally
{
    connection.Close();
}
}

public void SaveToDB(List<Region> regions)
{
    try
    {
        String query1 = "delete from region_resources_normalized;";
        MySQLCommand command = new MySQLCommand(query1, connection);
        connection.Open();
        command.ExecuteNonQuery();
    }
    catch (Exception e)
    {
        System.Windows.Forms.MessageBox.Show(e.Message);
    }
    finally
    {
        connection.Close();
    }
}

foreach (Region region in regions)
{
    try
    {
        String query2 = "insert into region_resources_normalized values (" + region.Id
            + ", " + region.SunEnergyNormalizedAmount.ToString().Replace(',', '.')
            + ", " + region.WindEnergyNormalizedAmount.ToString().Replace(',', '.')
            + ", " + region.GroundEnergyNormalizedAmount.ToString().Replace(',', '.')
            + ", " + region.WaterEnergyNormalizedAmount.ToString().Replace(',', '.')

```

```

        + ", " + region.TotalEnergyNormalizedAmount.ToString().Replace(',', '.')
        + " );";
        MySqlCommand command2 = new MySqlCommand(query2, connection);
        connection.Open();
        command2.ExecuteNonQuery();
    }
    catch (Exception e)
    {
        System.Windows.Forms.MessageBox.Show(e.Message);
    }
    finally
    {
        connection.Close();
    }
}

}

}

public partial class Form1 : Form
{
    DAO dao = new DAO();
    List<Region> regions;
    public Form1()
    {
        InitializeComponent();

        //Створення таблиці Region resources
        private void Form1_Load(object sender, EventArgs e)
        {
            dataGridView1.ColumnCount = 6;
            dataGridView1.Columns[0].Name = "№";
            dataGridView1.Columns[1].Name = "Область";
            dataGridView1.Columns[2].Name = "Показник сонячної радіації";
            dataGridView1.Columns[3].Name = "Показник середньої швидкості вітру";
            dataGridView1.Columns[4].Name = "Показник рівня енергії землі";
            dataGridView1.Columns[5].Name = "Показник гідроенергії";

            regions = dao.LoadFromDB();

            foreach (Region region in regions)
            {
                String[] row = new String[] {
                    region.Id.ToString(),
                    region.Name,
                    String.Format("{0:0.00}", region.SunEnergyAmount),
                    String.Format("{0:0.00}", region.WindEnergyAmount),
                    String.Format("{0:0.00}", region.GroundEnergyAmount),
                    String.Format("{0:0.00}", region.WaterEnergyAmount)
                };
                dataGridView1.Rows.Add(row);
            }
        }

        private void button1_Click(object sender, EventArgs e)
        {
            regions = new List<Region>();
            foreach (DataGridViewRow row in dataGridView1.Rows)
            {
                if (row.Cells[0].Value != null)

```

```

    {
        regions.Add(new Region(Int32.Parse(row.Cells[0].Value.ToString()),
            row.Cells[1].Value.ToString(),
            Double.Parse(row.Cells[2].Value.ToString()),
            Double.Parse(row.Cells[3].Value.ToString()),
            Double.Parse(row.Cells[4].Value.ToString()),
            Double.Parse(row.Cells[5].Value.ToString())));
    }
}

```

```

if (regions != null && regions.Count > 0)

```

```

{
    double minSun = regions[0].SunEnergyAmount;
    double maxSun = regions[0].SunEnergyAmount;
    double minWind = regions[0].WindEnergyAmount;
    double maxWind = regions[0].WindEnergyAmount;
    double minGround = regions[0].GroundEnergyAmount;
    double maxGround = regions[0].GroundEnergyAmount;
    double minWater = regions[0].WaterEnergyAmount;
    double maxWater = regions[0].WaterEnergyAmount;

```

```

foreach (Region region in regions)

```

```

{
    if (region.SunEnergyAmount > maxSun)
    {
        maxSun = region.SunEnergyAmount;
    }
    if (region.SunEnergyAmount < minSun)
    {
        minSun = region.SunEnergyAmount;
    }
    if (region.WindEnergyAmount > maxWind)
    {
        maxWind = region.WindEnergyAmount;
    }
    if (region.WindEnergyAmount < minWind)
    {
        minWind = region.WindEnergyAmount;
    }
    if (region.GroundEnergyAmount > maxGround)
    {
        maxGround = region.GroundEnergyAmount;
    }
    if (region.GroundEnergyAmount < minGround)
    {
        minGround = region.GroundEnergyAmount;
    }
    if (region.WaterEnergyAmount > maxWater)
    {
        maxWater = region.WaterEnergyAmount;
    }
    if (region.WaterEnergyAmount < minWater)
    {
        minWater = region.WaterEnergyAmount;
    }
}

```

```

foreach (Region region in regions)

```

```

{
    region.GroundEnergyNormalizedAmount = (region.GroundEnergyAmount - minGround) /
        (maxGround - minGround);
    region.WindEnergyNormalizedAmount = (region.WindEnergyAmount - minWind) /
        (maxWind - minWind);
}

```

```

        region.SunEnergyNormalizedAmount = (region.SunEnergyAmount - minSun) /
            (maxSun - minSun);
        region.WaterEnergyNormalizedAmount = (region.WaterEnergyAmount - minWater) /
            (maxWater - minWater);
        region.TotalEnergyNormalizedAmount = region.GroundEnergyNormalizedAmount +
            region.WindEnergyNormalizedAmount + region.SunEnergyNormalizedAmount +
            region.WaterEnergyNormalizedAmount;
    }

    dataGridView2.ColumnCount = 7;
    dataGridView2.Columns[0].Name = "№";
    dataGridView2.Columns[1].Name = "Область";
    dataGridView2.Columns[2].Name = "Питомий нормований потенціал сонячної енергії";
    dataGridView2.Columns[3].Name = "Питомий нормований потенціал вітрової енергії";
    dataGridView2.Columns[4].Name = "Питомий нормований потенціал енергії землі";
    dataGridView2.Columns[5].Name = "Питомий нормований потенціал гідроенергії";
    dataGridView2.Columns[6].Name = "Сумарний питомий нормований потенціал всіх видів енергії";
    dataGridView2.Rows.Clear();

    //створюємо таблицю Region resources normalized
    foreach (Region region in regions)
    {
        String[] row = new String[] {
            region.Id.ToString(),
            region.Name,
            String.Format("{0:0.0000}", region.SunEnergyNormalizedAmount),
            String.Format("{0:0.0000}", region.WindEnergyNormalizedAmount),
            String.Format("{0:0.0000}", region.GroundEnergyNormalizedAmount),
            String.Format("{0:0.0000}", region.WaterEnergyNormalizedAmount),
            String.Format("{0:0.0000}", region.TotalEnergyNormalizedAmount),
        };
        dataGridView2.Rows.Add(row);
    }

    dao.SaveToDB(regions);

    button2.Enabled = true;
}

private void button2_Click(object sender, EventArgs e)
{
    new MapForm(regions).Show();
}

private void DataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
}

}

private void MapForm_Load(object sender, EventArgs e)
{
}

private void PictureBox1_Paint(object sender, PaintEventArgs e)
{
    DrawDiagram(e, 510, 460, Regions[0]);
    DrawDiagram(e, 270, 260, Regions[1]);
    DrawDiagram(e, 80, 70, Regions[2]);
    DrawDiagram(e, 560, 260, Regions[3]);
}

```

```

DrawDiagram(e, 690, 270, Regions[4]);
DrawDiagram(e, 260, 110, Regions[5]);
DrawDiagram(e, 30, 300, Regions[6]);
DrawDiagram(e, 580, 370, Regions[7]);
DrawDiagram(e, 90, 270, Regions[8]);
DrawDiagram(e, 330, 110, Regions[9]);
DrawDiagram(e, 380, 290, Regions[10]);
DrawDiagram(e, 730, 220, Regions[11]);
DrawDiagram(e, 20, 210, Regions[12]);
DrawDiagram(e, 410, 380, Regions[13]);
DrawDiagram(e, 330, 430, Regions[14]);
DrawDiagram(e, 540, 210, Regions[15]);
DrawDiagram(e, 160, 70, Regions[16]);
DrawDiagram(e, 570, 130, Regions[17]);
DrawDiagram(e, 110, 220, Regions[18]);
DrawDiagram(e, 620, 200, Regions[19]);
DrawDiagram(e, 510, 380, Regions[20]);
DrawDiagram(e, 210, 230, Regions[21]);
DrawDiagram(e, 340, 240, Regions[22]);
DrawDiagram(e, 140, 300, Regions[23]);
DrawDiagram(e, 380, 60, Regions[23]);
}

```

//метод побудови стовпчастих діаграм

```

private void DrawDiagram(PaintEventArgs e, int x, int y, Region region)
{
    Brush brush = new SolidBrush(Color.Orange);
    // x, y, width, height
    e.Graphics.FillRectangle(brush, x, y, 10, (float)(20 * region.SunEnergyNormalizedAmount));
    brush = new SolidBrush(Color.Green);
    e.Graphics.FillRectangle(brush, x + 10, y, 10, (float)(20 * region.WindEnergyNormalizedAmount));
    brush = new SolidBrush(Color.Red);
    e.Graphics.FillRectangle(brush, x + 20, y, 10, (float)(20 * region.GroundEnergyNormalizedAmount));
    brush = new SolidBrush(Color.Blue);
    e.Graphics.FillRectangle(brush, x + 30, y, 10, (float)(20 * region.WaterEnergyNormalizedAmount));
    brush = new SolidBrush(Color.Black);
    e.Graphics.FillRectangle(brush, x + 40, y, 10, (float)(20 * region.TotalEnergyNormalizedAmount));
}
private void pictureBox1_Click(object sender, EventArgs e)
{
}

}
}

```



## ДОДАТОК 3

Автоматизована система оцінки ресурсів відновлюваних джерел енергії  
території з використанням показника сумарного питомого нормованого  
потенціалу

### ОПИС ПРОГРАМНОГО МОДУЛЮ

УКР.КПІм.ІгоряСікорського\_ТЕФ\_АПЕПС\_ТР5288\_19Б

Аркушів 9

Київ 2019

## АНОТАЦІЯ

Додаток містить опис системи, що розроблена для оцінки ресурсів відновлюваних джерел енергії території з використанням показника сумарного питомого нормованого потенціалу.

В додатку виконуються такі функції:

- введення вхідних даних для обчислень ;
- обчислення;
- виведення результатів обчислень.

Вхідні дані отримуються з попередньо створеної бази даних.

Вихідні дані виводяться на формі Windows Forms через елементи керування.

Система створена в Visual Studio Community 2017 з використанням мови програмування C#. Графічний інтерфейс користувача розроблений за допомогою API Windows Forms. Для збереження даних використана система управління базами даних (СУБД) MySQL Server. Для доступу до бази даних та автоматичного створення сутностей та управління ними був застосований Entity Framework 6.0.

## ЗМІСТ

ЗАГАЛЬНІ ВІДОМОСТІ .....	60
ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ .....	61
ОПИС ЛОГІЧНОЇ СТРУКТУРИ .....	62
ВИКОРИСТОВУВАНІ ТЕХНІЧНІ ЗАСОБИ.....	63
ВИКЛИК І ЗАВАНТАЖЕННЯ .....	64
ВХІДНІ І ВИХІДНІ ДАНІ .....	65

## ЗАГАЛЬНІ ВІДОМОСТІ

У цьому додатку описано систему оцінки відновлювальних джерел енергії території з використанням показника сумарного нормованого питомого потенціалу.

Модулі системи надано в ДОДАТКУ 2.

Розроблений додаток працює в операційних системах Windows7, Windows8, Windows10.

Компоненти необхідні для установки: MySQL Workbench, Microsoft SQL Server.

Використана мова програмування — C#.

## **ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ**

Розроблена система обчислює нормовані значення питомого потенціалу ресурсів ВДЕ для кожного регіону та показник сумарного питомого нормованого потенціалу забезпеченості областей ресурсами ВДЕ.

Розроблений програмний додаток рекомендований для використання спеціалістами енергетики, інженерами з проектування енергетичних процесів та систем, фахівцями підприємств з енергозабезпечення, а також зацікавленим спеціалістам інших профілів.

## ОПИС ЛОГІЧНОЇ СТРУКТУРИ

Програмний додаток розроблений за схемою дворівневої архітектури «Клієнт-Сервер». Доступ до бази даних здійснюється за допомогою Entity Framework.

На початку роботи, після запуску програмного додатку з'являється головне вікно форми. Далі необхідно натиснути кнопку “Обчислити нормовані показники”, після цього, з'явиться таблиця з обрахованими нормалізованими питомими показниками ресурсів ВДЕ та з обрахованим показником сумарного нормованого питомого потенціалу для кожного регіону. Після натиснення на кнопку “Мапа України”, у новому вікні буде зображено карту України із стовпчастими діаграмами показників для кожного регіону.

## **ВИКОРИСТОВУВАНІ ТЕХІЧНІ ЗАСОБИ**

Для розробки програмного продукту було обрано:

- операційну систему Windows 10;
- об'єктно-орієнтовану мову програмування C#;
- середовище MS Visual Studio 2017;
- фреймворк ADO.NET Entity Framework;
- систему керування базами даних MySQL Server.

## **ВИКЛИК І ЗАВАНТАЖЕННЯ**

Для запуску розробленого програмного додатку необхідно запустити виконуваний файл UAEnergy .exe, інсталяція системи не потрібна.

Після запуску програми користувач отримує доступ у головну форму, з якої може починати роботу в додатку та виконувати необхідні обчислення.



## **ВХІДНІ І ВИХІДНІ ДАНІ**

Вхідні дані мають вигляд цілого числа або числа з плаваючою комою.

Вхідні дані для роботи програмного додатку зчитуються з таблиці Region recourse бази даних.

Вихідні дані мають вигляд числа з плаваючою комою та виводяться на формі Windows Forms через елементи керування

Вихідні дані, які отримуються в результаті роботи програми є нормовані показники ресурсів відновлювальних джерел енергії та показник сумарного питомого нормованого потегціалу.